

В этом посте речь пойдёт о функции `printf()`, с которой нам уже приходилось сталкиваться ещё в самой первой программе. Как отмечалось ранее, название функции - сокращение от **print formatted** (форматированный вывод)

А само определение выглядит следующим образом:

```
int printf(const * char format, ...);
```

language-c

Первый параметр `format` – это указатель на строку. Пока его можно воспринимать просто как строку, в которой определен формат для вывода информации в стандартный поток `stdout`. А троеточие определяет произвольное число дополнительных параметров. Обычно, это переменные или выражения, значения которых следует выводить в заданном формате. В качестве возвращаемого типа указан `int`, то есть функция возвращает целое число. Это число соответствует количеству выведенных символов в стандартный поток `stdout` (в нашем случае на экран). Обычно, на практике, этим значением пренебрегают и вызывают функцию `printf()`, как говорят, **ради побочного эффекта**, т.е. ради передачи данных в выходной поток.

В первой программе мы использовали `printf()` для вывода строки "Hello, world!". Но она способна на большее! Сейчас мы рассмотрим лишь основные её возможности, которые наиболее часто применяются на практике. Если вам понадобится больше информации, то справочные руководства всегда к вашим услугам.

Итак, первый параметр `format` функции `printf()` определяет не просто выводимую строку, а формат выдаваемых данных. И для этого в нашем распоряжении имеются, так называемые, **спецификаторы преобразования**. Основные из них следующие.

Спецификатор	Описание
%d или %i	Целое число со знаком в десятичной форме.
%u	Целое беззнаковое (unsigned) число в десятичной форме.
%o	Целое беззнаковое (unsigned) число в восьмеричной форме.
%x или %X	Целое беззнаковое (unsigned) число в шестнадцатеричной форме.
%f или %F	Вещественное число в виде десятичной дроби.
%e или %E	Вещественное число в экспоненциальной форме.
%c	Символ в соответствии с текущей кодовой таблицей.

%s	Строка (последовательность символов).
%%	Запись символа % в форматной строке.

Существуют и другие спецификаторы, но мы остановимся на этих (+ спецификатор %s мы рассмотрим, когда будем рассматривать строки)

Напишем небольшую программу - пример:

```
#include <stdio.h>
```

language-c

```
int main(){  
    int var = 1208;  
    printf()  
}
```